<u>IN THE CLAIMS</u>

Please amend claims 2, 3, 10, and 36 as set forth below. Non-amended claims are

included for the convenience of the Examiner.

1.   (Original) A client comprising:

a memory having an application and a data structure stored therein, wherein the data

structure identifies positions of the compressed codestream on a server and identifies data of

the compressed codestream already buffered at the client, if any;

a processor coupled to the memory to execute the application to generate a request for

portions of the compressed codestream based on indications of which portions of the

codestream are already stored in the memory as indicated by the data structure.

2.   (Currently Amended) The client defined in Claim 1 wherein the processor ~~creates~~

~~a new codestream by integrating~~, prior to decoding, integrates previously obtained portions of

the compressed codestream with~~in~~ portions of the compressed codestream received as a result of

the request to create a new codestream by putting packets in the order the packets appeared in the

compressed codestream and by updating codestream markers to reflect that the previously

obtained portions of the compressed codestream and the portions of the compressed codestream

received as a result of the request are part of the new codestream~~, and the processor updates~~

~~markers for the new codestream~~.

3.   (Currently Amended) A system comprising:

a server to store a compressed codestream corresponding to image data; and

a client coupled to the server via a network environment, wherein the client includes a

memory having an application and a data structure stored therein, wherein the data structure

identifies positions of the compressed codestream on the server and identifies data of the

compressed codestream already buffered at the client, if any, and further wherein the client

operates to requests bytes of the compressed codestream from the server that are not already

stored in the memory and generates a new codestream from ~~decoded image data requested by a~~

~~user~~ from the bytes of the compressed codestream requested from the server and any portion of

the compressed codestream previously stored in the memory necessary to create the image data,

the new codestream generated by putting packets in the order the packets appeared in the

compressed codestream and by updating codestream markers to reflect that the bytes of the

compressed codestream requested from the server and any portion of the compressed codestream

previously stored in the memory necessary to create the image data are part of the new

codesteram, the client to generate image data by decoding the new codestream.

4.   (Original) The system defined in Claim 3 wherein the portions of the compressed

codestream are selected from a group consisting of packets, tile parts, and coded data segments

from a codebook.

5.   (Original) The system defined in Claim 3 wherein, when executing the

application, the client

        determines image characteristics that a user requests,

        selects data of a compressed codestream that corresponds to the image characteristics,

        determines data of a compressed codestream that corresponds to the image characteristics

that is not already buffered at the client,

issues requests to the server to obtain the data of a compressed codestream that

corresponds to the image characteristics that is not already buffered at the client,

integrates data received from the server with any previously buffered data of the

compressed codestream that corresponds to the image characteristics,

decodes the data of the compressed codestream that corresponds to the image

characteristics, and

displays an image corresponding to the decoded compressed codestream.

6.   (Original) The system defined in Claim 3 wherein the server serves byte requests.

7.   (Original) The system defined in Claim 3 wherein the client further comprises a

software decoder, and the client creates the compressed codestream for the software decoder by

integrating bytes requested with previously obtained bytes.

8.   (Original) The system defined in Claim 3 wherein the client determines the

location and length of each packet.

9.   (Original) The system defined in Claim 8 wherein the client requests a

headerlength of a compressed file from the server that includes one or more file format boxes

and a main header of the codestream box from which the client determines the location and

length of each packet.

10.   (Currently Amended) The system defined in Claim 9 wherein the main header

includes two marker segments indicative of a ~~byte~~ map to every packet.

11.   (Original) The system defined in Claim 10 wherein the two marker segments comprise the TLM and PLM marker segments.

12.   (Original) The system defined in Claim 9 wherein the server comprises a script that streams only requested bytes to the client that are in a range specified in the request.

13.   (Original) The system defined in Claim 3 wherein the compressed codestream comprises a JPEG 2000 codestream.

14.   (Currently Amended) A method for processing image data by a client, the method comprising:

determining image characteristics that a user requests;

selecting data of a compressed codestream that corresponds to the image characteristics;

determining data of a compressed codestream that corresponds to the image characteristics that is not already buffered at the client, wherein determining the data comprises using a data structure that identifies positions of portions of the compressed codestream on a server and that identifies data of the compressed codestream already buffered at the client;

issuing requests to the server to obtain the data of a compressed codestream that corresponds to the image characteristics that is not already buffered at the client;

integrating data received from the server with any previously buffered data of the compressed codestream that corresponds to the image characteristics to create a new codestream having markers that reflect both data received from the server and any previously buffered data

of the compressed codestream that corresponds to the image characteristics are included in the

new codestream;

decoding the data of the compressed codestream that corresponds to the image

characteristics; and

displaying an image corresponding to the decoded compressed codestream.

15.   (Original) The method defined in Claim 14 further comprising the server serving

byte requests.

16.   (Original) The method defined in Claim 14 further comprising compiling the

compressed codestream for a software decoder on the client.

17.   (Original) The method defined in Claim 14 further comprising determining the

location and length of each packet.

18.   (Original) The method defined in Claim 17 further comprising requesting a

headerlength of a compressed file from the server that includes one or more file format boxes

and a main header of the codestream box from which the client determines the location and

length of each packet.

19.   (Original) The method defined in Claim 18 wherein the main header includes two

marker segments indicative of a byte map to every packet.

20.   (Original) The method defined in Claim 19 wherein the two marker segments

comprise the TLM and PLM marker segments.

21.   (Original) The method defined in Claim 14 wherein the compressed codestream

comprises a JPEG 2000 codestream.

22.   (Currently Amended) An article of manufacture having one or more recordable

media having executable instructions stored thereon which, when executed by the system cause

the system to:

indent determine image characteristics that a user requests;

select data of a compressed codestream that corresponds to the image characteristics;

determine data of a compressed codestream that corresponds to the image characteristics

that is not already buffered at the client, wherein determining the data comprises using a data

structure that identifies positions of portions of the compressed codestream on a server and that

identifies data of the compressed codestream already buffered at the client;

issue requests to the server to obtain the data of a compressed codestream that

corresponds to the image characteristics that is not already buffered at the client;

integrate data received from the server with any previously buffered data of the

compressed codestream that corresponds to the image characteristics to create a new codestream

having markers that reflect both data received from the server and any previously buffered data

of the compressed codestream that corresponds to the image characteristics are included in the

new codestream;

decode the data of the compressed codestream that corresponds to the image

characteristics; and

display an image corresponding to the decoded compressed codestream.

23.   (Original) The article of manufacture defined in Claim 22 further comprising instructions which, when executed by the system, cause the system to create the compressed codestream for a software decoder on the client.

24.   (Original) The article of manufacture defined in Claim 22 further comprising instructions which, when executed by the system, cause the system to determine the location and length of each packet.

25.   (Original) The article of manufacture defined in Claim 24 further comprising instructions which, when executed by the system, cause the system to request a headerlength of a compressed file from the server that includes one or more file format boxes and a main header of the codestream box from which the client determines the location and length of each packet.

26.   (Original) The article of manufacture defined in Claim 25 wherein the main header includes two marker segments indicative of a byte map to every packet.

27.   (Original) The article of manufacture defined in Claim 26 wherein the two marker segments comprise the TLM and PLM marker segments.

28.   (Original) The article of manufacture defined in Claim 22 wherein the compressed codestream comprises a JPEG 2000 codestream.

29.   (Original) A method comprising:

receiving requests for a codestream for a JPEG 2000 image;

converting requested information to JPEG DCT compression; and

sending converted requested information using the Internet Imaging Protocol (IIP).

30.   (Original) A method comprising:

providing information to enable a client to determine image size and a maximum number

of resolutions;

receiving a request for one or more tiles of a codestream of compressed data; and

parsing the codestream of compressed data to locate packets related to the request,

including packets for a requested tile at a resolution less than or equal to the resolution requested

and all layers.

31.   (Original) The method defined in Claim 30 wherein packets located as a result of

parsing includes all packets for a requested tile at a resolution less than or equal to a resolution

requested and all layers.

32.   (Original) The method defined in Claim 30 where the information comprises an

HTML document.

33.   (Original) The method defined in Claim 30 where the information comprises an

XML document.

34.   (Original) A system comprising:

a client

a server coupled to the client via a network environment, the server to store a compressed

codestream corresponding to image data, wherein the server provides information to enable a

client to determine image size and a maximum number of resolutions, receives a request for one

or more tiles of a codestream of compressed data, and parses the codestream of compressed data

to locate packets related to the request.

35.   (Original) The system defined in Claim 34 wherein the request system includes

component, layer and precinct information.

36.   (Currently Amended) The system defined in Claim 34 wherein the request syntax

is as follows:

TIL = resolution, component, layer, precinct, tile

wherein resolution, component, layer, precinct, tile are defined by the JPEG 2000 Standard.

37.   (Original) The system defined in Claim 34 wherein the layer is a JPEG 2000

layer and the precinct is a JPEG 2000 precinct.

38.   (Original) The system defined in Claim 34 wherein one or more variables in the

request syntax is represented using a wildcard.